UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/761,404	01/16/2001	Bjarne Steensgaard	MS 158288.1/40062.119US01	4797

23552 7590 02/01/2006

MERCHANT & GOULD PC
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 02/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/761,404	STEENSGAARD, BJARNE	
	Examiner	Art Unit	
	Syed J. Ali	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 November 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-51* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-51* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed November 1, 2005. Claims 1-51 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 102

3. **Claims 1-20, 25-30, 33-38, 40-43, and 47-50 are rejected under 35 U.S.C. 102(e) as being anticipated by Blais et al. (USPN 6,505,344) (hereinafter Blais).**

4. As per claims 1, 25, 34-35, 40, and 47, Blais teaches the invention as claimed, including a computer program product stored on at least one physical computer readable media and encoding a computer program for executing on a computer system a computer implemented method for managing allocation of program data in a target program between one or more thread-specific heaps and at least one shared heap, the program data including thread-specific data and shared data, the computer implemented method comprising:

analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread and the shared data (col. 9 lines 3-20);

configuring the target program to allocate the thread-specific data of the first program thread to a first thread-specific heap, responsive to the analyzing operation (col. 11 lines 44-47, 51-57); and

configuring the target program to allocate the shared data to the shared heap, responsive to the analyzing operation (col. 11 lines 47-50, 57-58).

5. As per claim 2, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the analyzing operation comprises analyzing the target program to distinguish among the thread-specific data of the first program thread, the thread-specific data of a second program thread, and the shared data, and wherein the computer implemented method further comprises:

configuring the target program to allocate the thread-specific data of the second program thread to a second thread-specific heap, responsive to the analyzing operation (Fig. 9, step 850).

6. As per claim 3, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the analyzing operation comprises:

identifying program data in the target program as the thread-specific data of the first program thread, if the program data is not referenced by any other program thread of the target program (col. 11 lines 43-47).

7. As per claim 4, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the analyzing operation comprises:

identifying program data in the target program as the thread-specific data of the first program thread based on a thread escape analysis (col. 9 lines 3-20).

8. As per claims 5 and 26, Blais teaches the invention as claimed, including the computer program product of claims 1 and 25, wherein the target program further includes a second program thread and the analyzing operation comprises:

identifying program data in the target program as the shared data, if the program data is referenced by the first program thread and the second program thread of the target program (col. 11 lines 47-50, 57-58).

9. As per claim 6, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the analyzing operation occurs prior to the execution of the target program (col. 9 lines 3-20).

10. As per claims 7 and 27, Blais teaches the invention as claimed, including the computer program product of claims 1 and 25, wherein the operation of configuring the target program to allocate the thread-specific data comprises:

replacing an original allocation instruction in the target program with a new instruction that allocates the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread (col. 11 lines 44-47, 51-57).

11. As per claims 8 and 28, Blais teaches the invention as claimed, including the computer program product of claims 1 and 25, wherein the operation of configuring the target program to allocate the thread-specific data comprises:

leaving an original allocation instruction in the target program to allocate the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread (col. 11 lines 44-47, 51-57).

12. As per claim 9 and 30, Blais teaches the invention as claimed, including the computer program product of claims 1 and 25, wherein the operation of configuring the target program to allocate the shared data comprises:

leaving an original allocation instruction in the target program to allocate the shared data to the shared heap (col. 11 lines 47-50, 57-58).

13. As per claims 10 and 29, Blais teaches the invention as claimed, including the computer program product of claims 1 and 25, wherein the operation of configuring the target program to allocate the shared data comprises:

replacing an original allocation instruction in the target program with a new instruction that allocates the shared data to the shared heap (col. 11 lines 47-50, 57-58).

14. As per claims 11 and 33, Blais teaches the invention as claimed, including the computer program product of claims 1 and 35, wherein the operation of configuring the target program to allocate the thread-specific data comprises:

configuring an allocation parameter associated with the thread-specific data indicating that the thread-specific data of the first program thread is to be allocated in the one of the thread-specific heaps (col. 11 lines 44-47, 51-57).

15. As per claim 12, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the operation of configuring the target program to allocate the thread-specific data further comprises:

allocating the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread, responsive to an allocation parameter (col. 11 lines 44-47, 51-57).

16. As per claim 13, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data comprises:

configuring an allocation parameter associated with the shared data indicating that the shared data is to be allocated in the shared heap (col. 11 lines 47-50, 57-58).

17. As per claim 14, Blais teaches the invention as claimed, including the computer program product of claim 13 wherein the operation of configuring the target program to allocate the shared data further comprises:

allocating the shared data to the shared heap, responsive to the allocation parameter (col. 11 lines 47-50, 57-58).

18. As per claim 15, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the operation of configuring the target program to allocate the thread-specific data occurs prior to execution of the target program (col. 9 lines 3-20).

19. As per claim 16, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data occurs prior to execution of the target program (col. 9 lines 3-20).

20. As per claims 17, 36, 41, and 48, Blais teaches the invention as claimed, including the computer program of claims 1, 35, 40, and 47, wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of garbage collection of the shared data in the shared heap (col. 1 lines 63-67; col. 2 lines 5-11, 30-39).

21. As per claim 18, Blais teaches the invention as claimed, including the computer program of claim 1 wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of garbage collection of a second thread-specific heap (col. 1 lines 63-67; col. 2 lines 5-11, 30-39).

22. As per claims 19, 37, 42, and 49, Blais teaches the invention as claimed, including the computer program of claims 1, 35, 40, and 47, wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of the execution of another program thread in the target program (col. 1 lines 63-67; col. 2 lines 5-11, 30-39).

23. As per claims 20, 38, 43, and 50, Blais teaches the invention as claimed, including the computer program of claims 1, 35, 40, and 47 wherein the computer implemented method further comprises:

garbage collecting the shared data from the shared heap independently of garbage collection of the thread-specific data in the first thread-specific heap (col. 1 lines 63-67; col. 2 lines 5-11, 30-39).

Claim Rejections - 35 USC § 103

24. **Claims 21-24, 31-32, 39, 44-46, and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blais in view of Pinter et al. (USPN 6,457,023) (hereinafter Pinter).**

25. As per claims 21, 39, 44, and 51, Pinter teaches the invention as claimed, including the computer program of claims 1, 35, 40, and 47, wherein the computer implemented method further comprises:

maintaining a remembered set identifying references to one or more shared data in the shared heap (col. 5 lines 15-19; col. 6 lines 55-65; col. 7 lines 3-13, 41-56).

26. Blais teaches the invention as claimed, including collecting the shared heap independently of garbage collection of the first thread-specific heap (col. 1 lines 63-67; col. 2 lines 5-11, 30-39).

27. It would have been obvious to one of ordinary skill in the art to combine Blais and Pinter because generational garbage collectors suffer from runtime overhead due to an inability to identify the lifetime of objects. This deficiency, which is noted by Applicant, is somewhat rectified by escape analysis, which is commonly used to determine the reachability of methods and threads in addition to determining the "lifetime" of an object (Blais, col. 2 lines 30-39). Pinter remedies the problems associated with generational garbage collectors by providing an estimate of object lifetime by performing pointer analysis. The pointer analysis also performs reachability analysis to identify if an object is accessed by a single thread or multiple threads (col. 2 lines 58-65). This pointer analysis proves that a particular object is accessed by a single thread, therefore being permissible to allocate on a local heap (col. 1 lines 7-11). Pinter points out that such data flow analysis can be used to optimize compilers (col. 1 line 65 - col. 2 line 6), while Blais performs compile-time analysis that determines which objects are local or shared.

28. As per claims 22, 31, and 45, Pinter teaches the invention as claimed, including the computer program product of claims 1, 25, and 40, wherein the computer implemented method further comprises:

collecting a portion of the shared data from the shared heap to leave an uncollected portion of the shared data in the shared heap, the uncollected portion of the shared data including shared data that is referenced by thread-specific data of the first thread-specific heap that has not yet been scanned (col. 5 lines 15-19; col. 6 lines 55-65; col. 7 lines 3-13, 41-56);

scanning the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data (col. 5 lines 15-19; col. 6 lines 55-65; col. 7 lines 3-13, 41-56); and

collecting the uncollected portion of the shared data from the shared heap, responsive to the scanning operation (col. 5 lines 15-19; col. 6 lines 55-65; col. 7 lines 3-13, 41-56).

29. As per claims 23, 32, and 46, Pinter teaches the invention as claimed, including the computer program product of claims 22, 31, and 40, wherein the computer implemented method further comprises:

collecting the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data (col. 5 lines 15-19; col. 6 lines 55-65; col. 7 lines 3-13, 41-56).

30. As per claim 24, Blais teaches the invention as claimed, including the computer program product of claim 1 wherein the shared heap is shared by a subset of the program threads of the target program, wherein the subset of program threads includes less than all of the program threads of the target program (col. 11 lines 44-58).

Response to Arguments

31. **Applicant's arguments filed November 1, 2005 have been fully considered but they are not persuasive.**

32. Applicant's arguments center on the premise that the claimed invention is distinct from Blais by virtue of the discrepancies between the claimed "thread-specific heap" and the "inlined invocation stack frame" of Blais. Specifically, Blais teaches allocating proven thread-specific data onto a stack frame that is local to a particular thread and automatically reclaiming the allocated data when it goes out of scope. On the other hand, the claimed invention allocates the proven thread-specific data to a local heap, which may be preferred to allocation on a stack, thereby allowing garbage collection of the thread-specific heap without impacting the performance of other threads.

33. While this distinction is not without merit, Examiner notes that the distinction is one that is contemplated by the prior art. Blais builds upon the thread escape analysis method developed by Choi et al. (see the "Escape Analysis for Java" document attached to the Office action mailed July 27, 2005). A person having ordinary skill in the art would recognize that Blais incorporates the teachings of Choi, including disclosures of known equivalency. Reference is hereby made to Choi et al. (USPN 6,381,738), which discloses the same escape analysis method described in the previously cited Choi document. Choi notes that allocating thread-specific data to an invocation stack frame is beneficial because it avoids garbage collection overhead by allowing the storage to be automatically reclaimed when the procedure returns (col. 1 lines 41-49). However, a programmer may on occasion prefer to allocate data to the heap rather than the stack, and a

region-based approach can be used that allows allocation of objects to a bound region in the heap, thereby alleviating the garbage collection overhead associated with collecting the entire heap (col. 1 line 65 - col. 2 line 5). Choi notes that the "region-based approach shall be regarded as equivalent to performing stack allocation of data" (col. 2 lines 3-5). Although there are some slight differences between heap and stack allocation, which are highlighted by Applicant, Choi indicates that allocation to a thread-specific heap is an alternative to stack allocation, and the same techniques of thread escape analysis to identify local data that can be garbage collected separately are applicable to either stack or heap based allocation. Accordingly, the claimed invention is merely a known equivalent, recognized in the prior art, of the stack-based allocation method described in Choi and refined by Blais.

Conclusion

34. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J. Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T. An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
January 30, 2006

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100